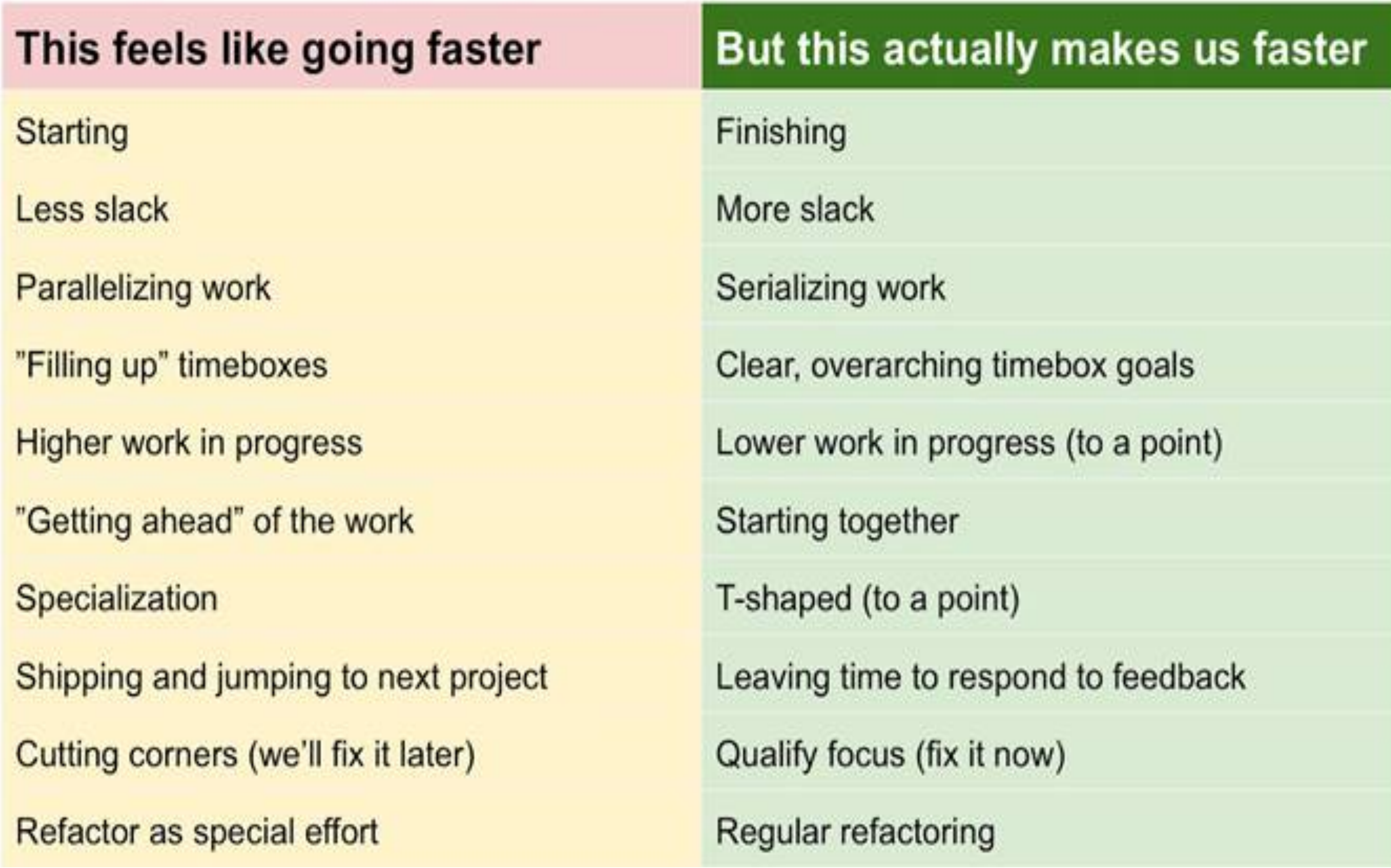
THE SCALING DILEMMA

Is there one “best” way?

LET’S CREATE A FOUNDATION FOR DISCUSSION



This feels like going faster	But this actually makes us faster
Starting	Finishing
Less slack	More slack
Parallelizing work	Serializing work
"Filling up" timeboxes	Clear, overarching timebox goals
Higher work in progress	Lower work in progress (to a point)
"Getting ahead" of the work	Starting together
Specialization	T-shaped (to a point)
Shipping and jumping to next project	Leaving time to respond to feedback
Cutting corners (we'll fix it later)	Qualify focus (fix it now)
Refactor as special effort	Regular refactoring

Source: John Cutler on Twitter

Handing off to test. Starting new work	Pairing with test. Finishing together
Hiring more people	Tooling, infrastructure, environments, quality
Throwing new team members into fray	Careful and safe onboarding
Individual assignments	Team goals, pairing, swarming, mobbing
Chase efficiency	Encourage messy but effective collaboration
Hands ON keyboards immediately	Team conducts research together
Larger batches (“less resetting”)	Smaller batches
Keep everyone heads down	Quiet/focused time punctuated by collab
Design then build	Participatory design, UX/dev pairing
Shared ops	Embedded ops (w/ some shared services)
Output velocity	Impact velocity
Stakeholder reviews and approvals	Partners in the trenches, regular participation
"Find a way around the blocker"	Swarm on the blocker

Being Agile vs Doing Agile

AGILE MINDSET

FRAMEWORKS

- Scrum
- Kanban
- XP
- SAFe
- Scrum@Scale

PRACTICES

- Daily Standup
- Sprint Planning
- Continuous Integration
- Retrospective
- Demos
- Test Driven Design

Organic Agile Evolution

Shu:

In this beginning stage the student follows the teachings of one master precisely. He concentrates on how to do the task, without worrying too much about the underlying theory. If there are multiple variations on how to do the task, he concentrates on just the one way his master teaches him.

Ha:

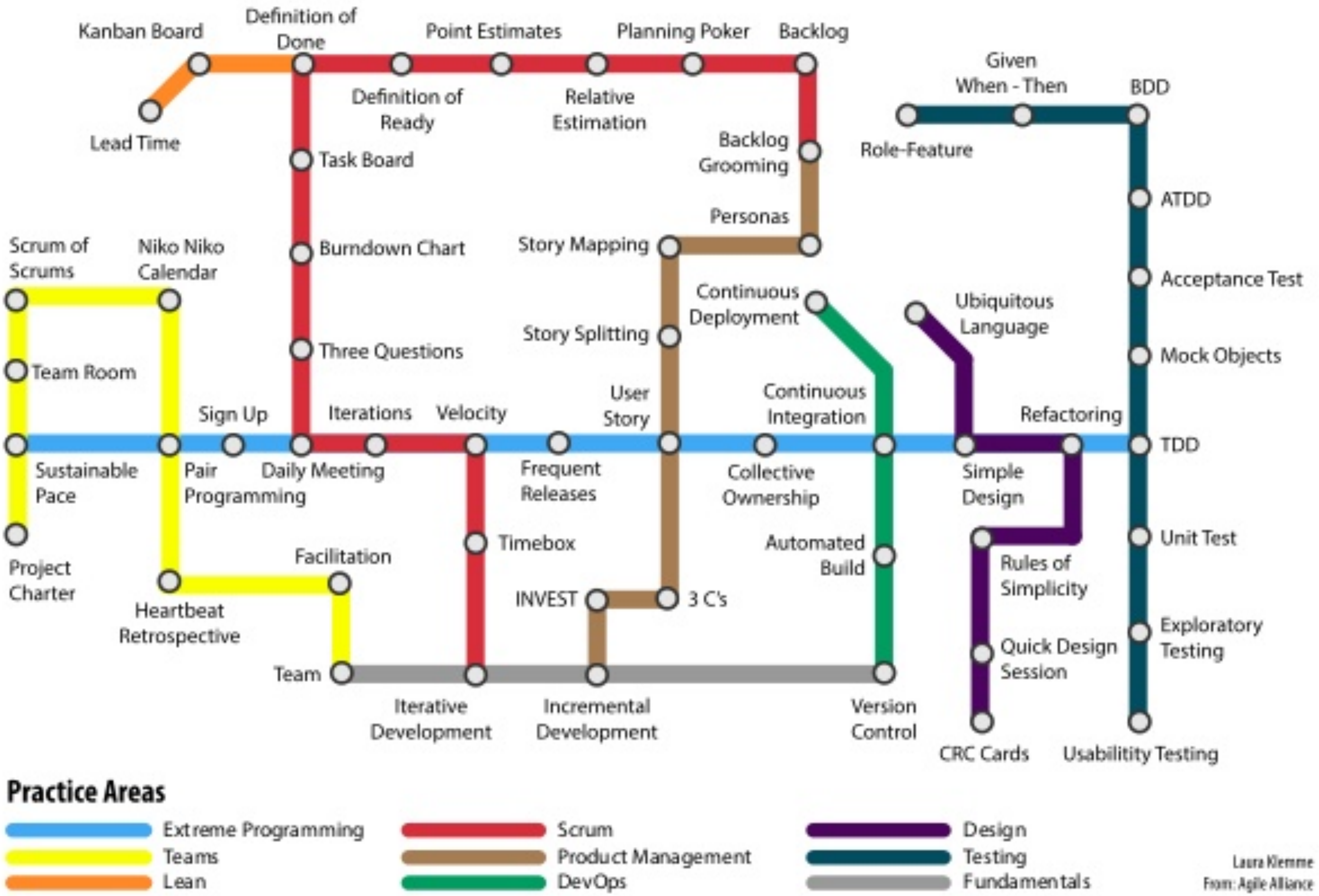
At this point the student begins to branch out. With the basic practices working he now starts to learn the underlying principles and theory behind the technique. He also starts learning from other masters and integrates that learning into his practice.

Ri:

Now the student isn't learning from other people, but from his own practice. He creates his own approaches and adapts what he's learned to his own particular circumstances.

Source: Martin Fowler 22 Aug 2014 Blog

Agile Tribes



Laura Klemme
From: Agile Alliance

WHAT ARE THE BENEFITS OF BEING AGILE?

Characteristics of agile teams

1. Higher quality – product works as expected
2. Customer satisfaction – product is what customer wants
3. Faster ROI
4. Early risk reduction

Do these qualities scale?

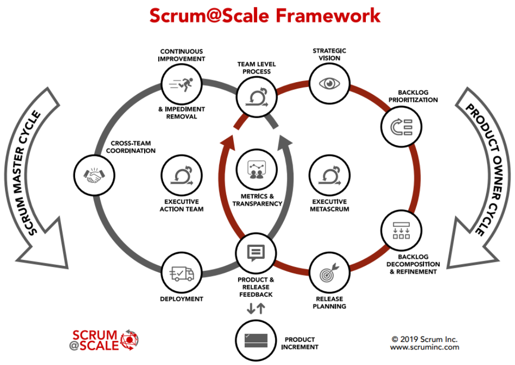
Scaled Agile

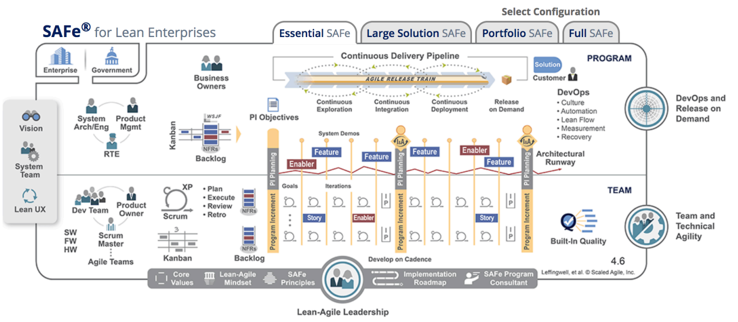
We still want:

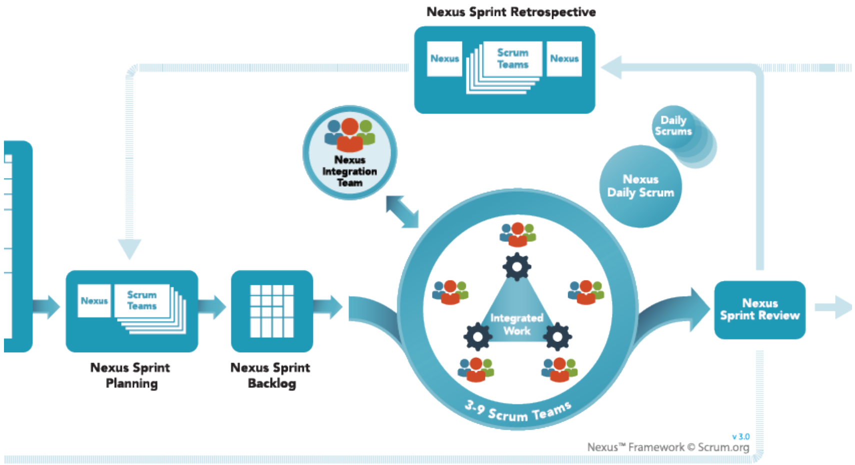
1. Higher quality – product works as expected
2. Customer satisfaction – product is what customer wants
3. Faster ROI
4. Early risk reduction

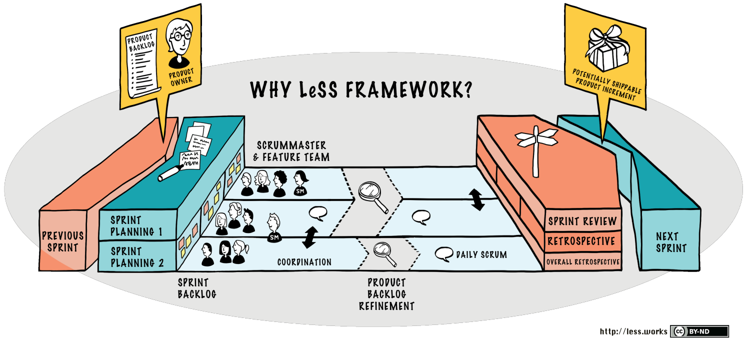
What if we apply Shu Ha Ri to the way we scale?

Some scaling frameworks

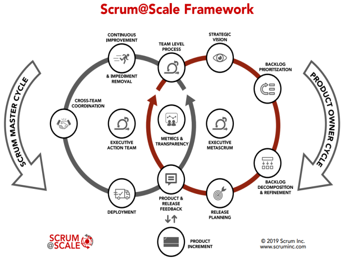


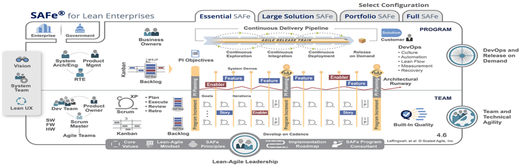


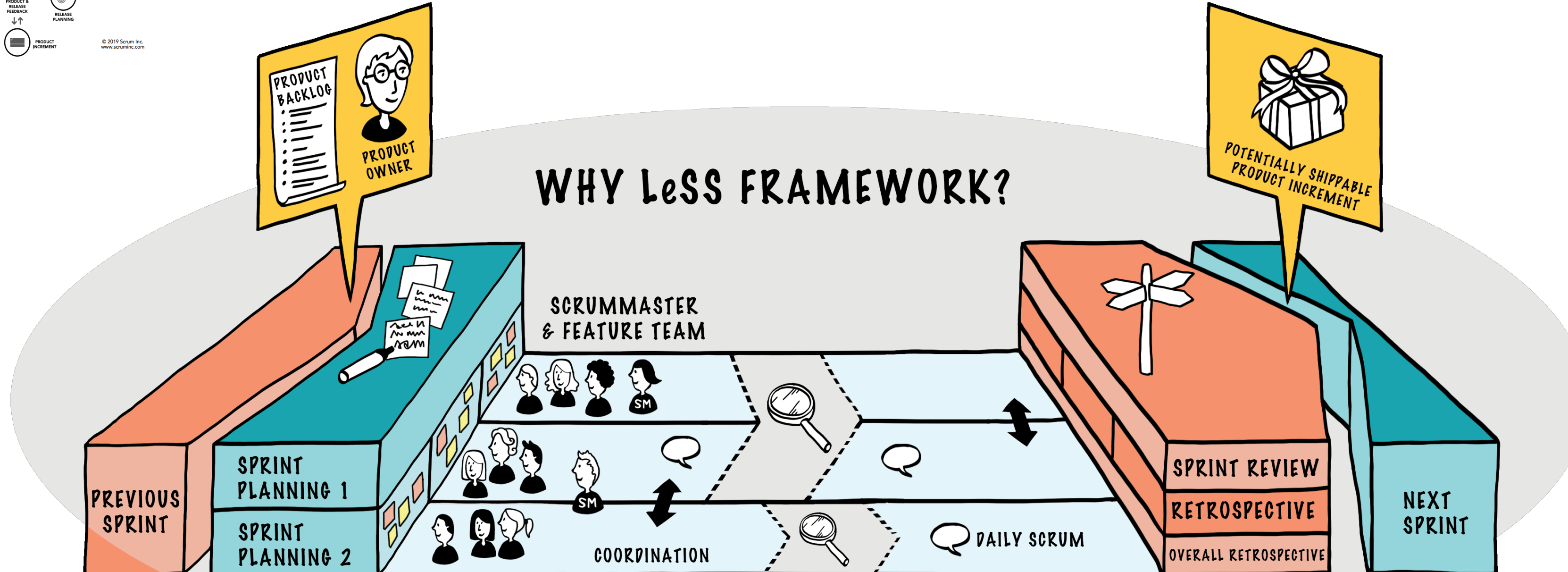




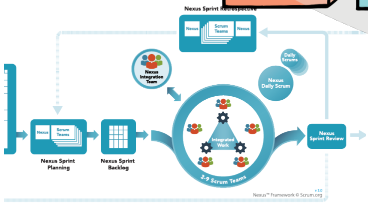
LeSS – Craig Larman & Bas Vodde



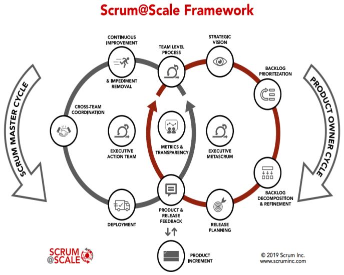


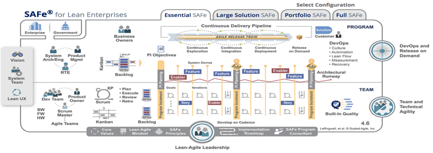


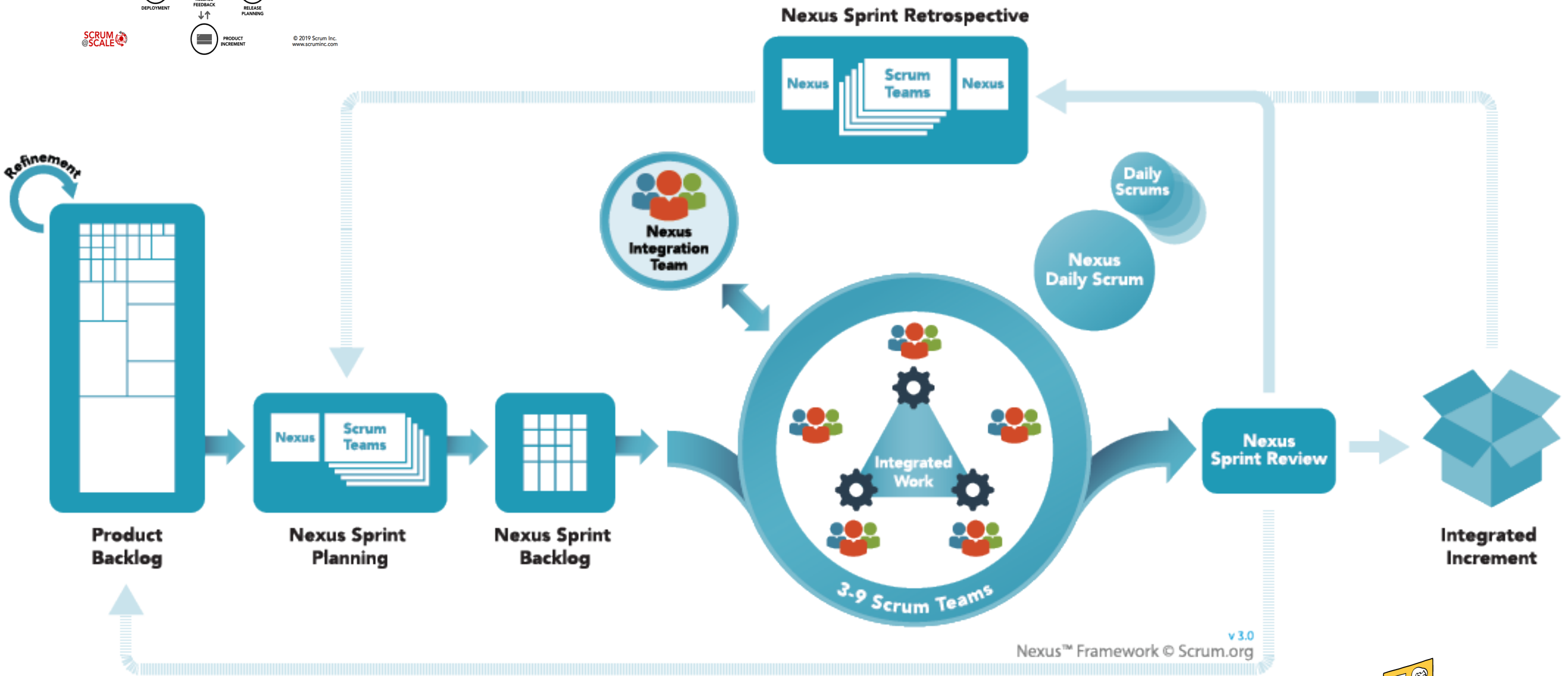
http://less.works (cc) BY-ND

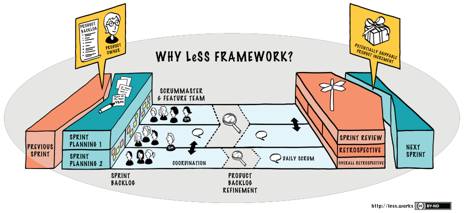


Nexus - Ken Schwaber

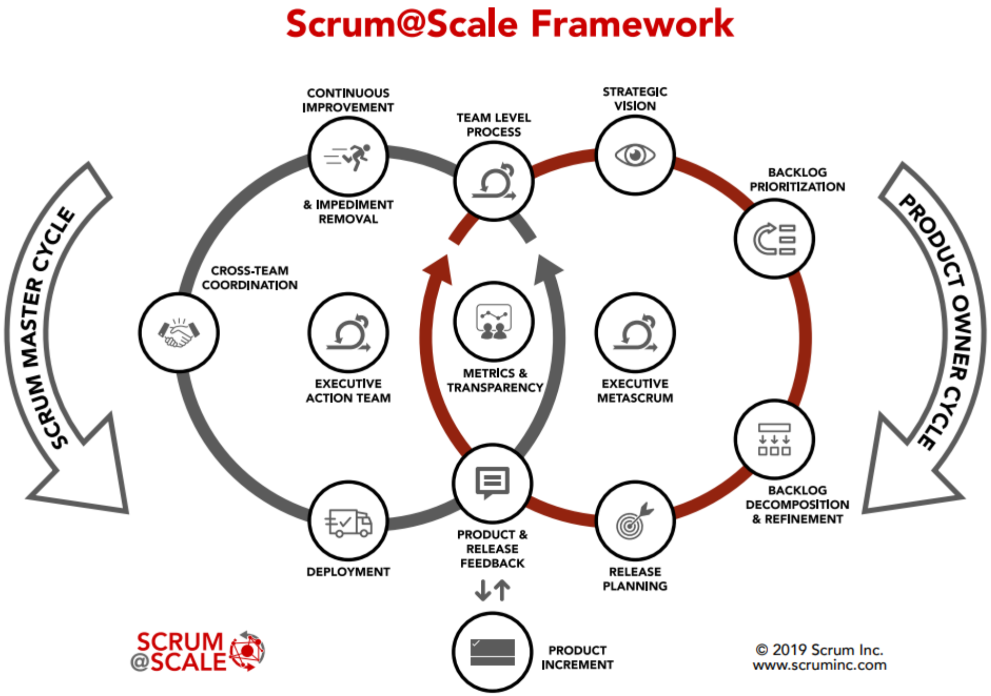


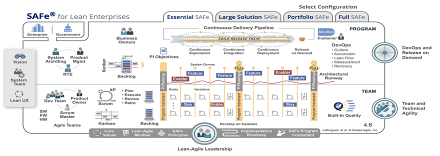


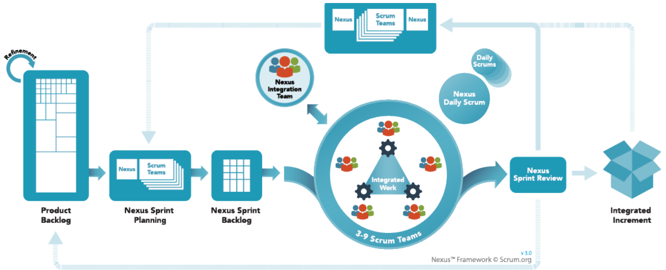


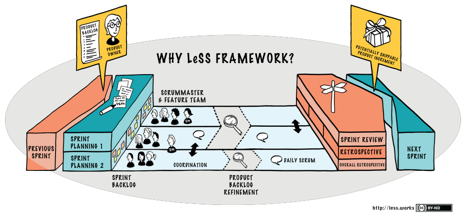


Scrum@Scale – Jeff Sutherland

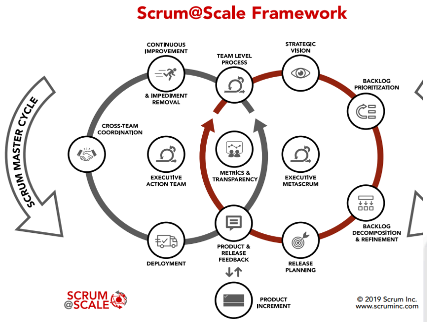


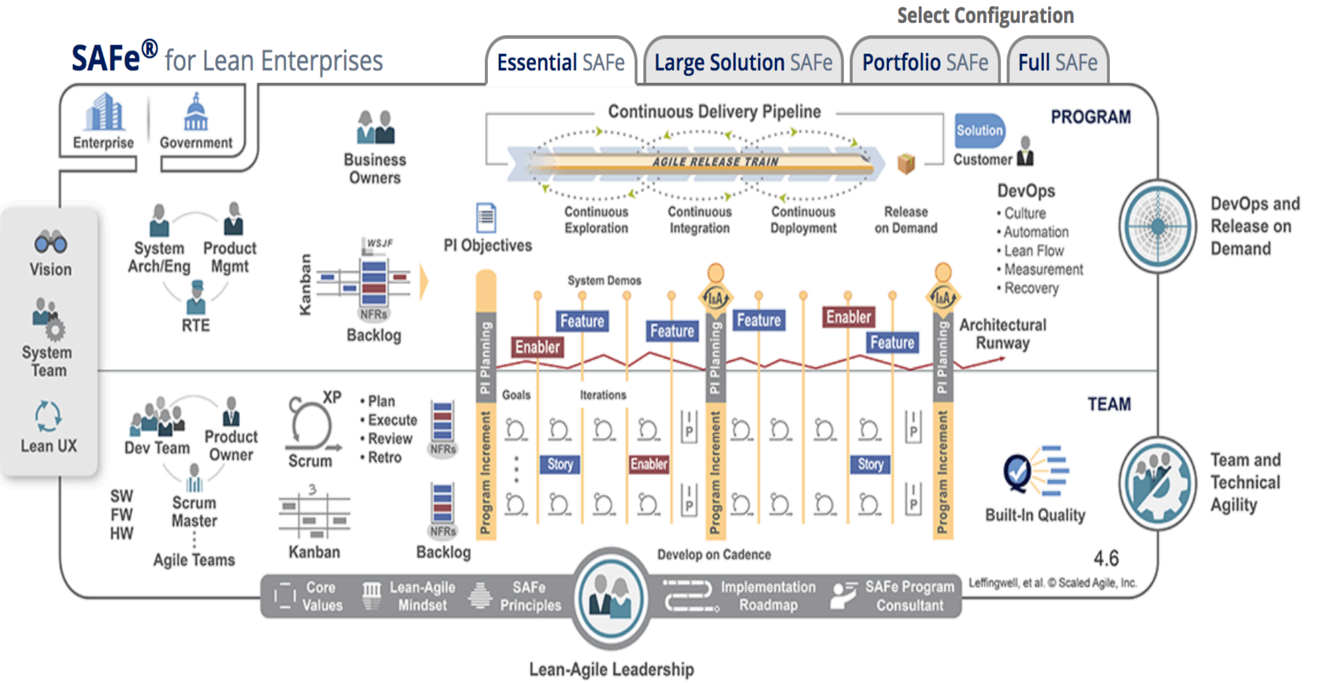


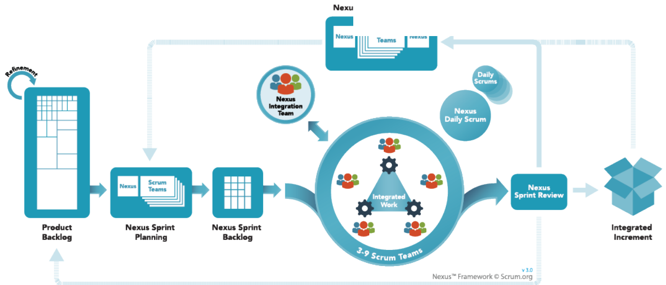


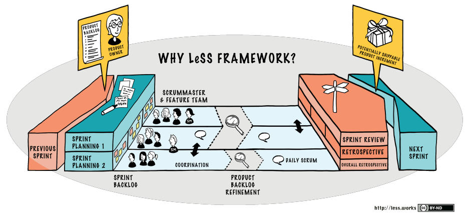


SAFe– Dean Leffingwell









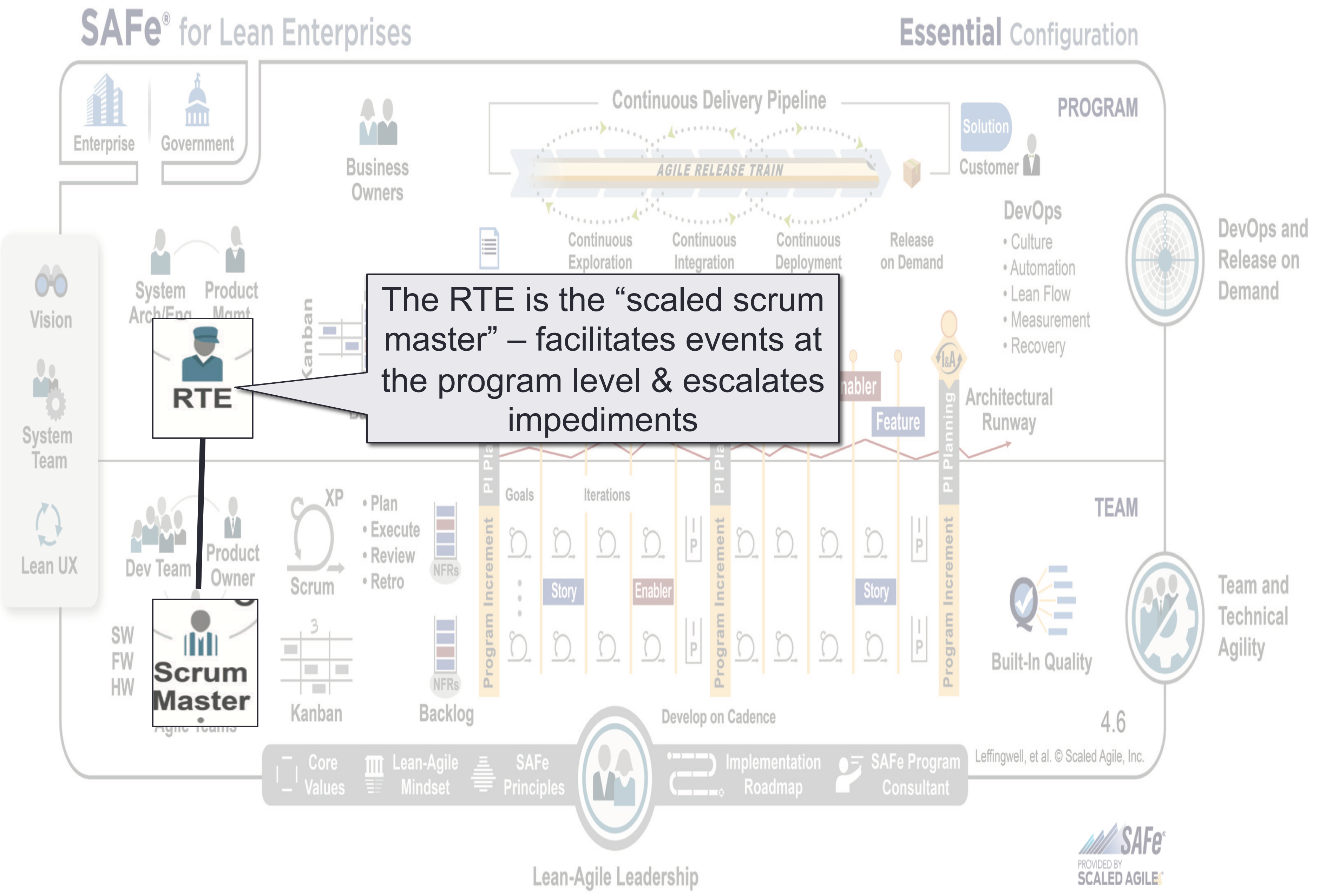
Caution before scaling

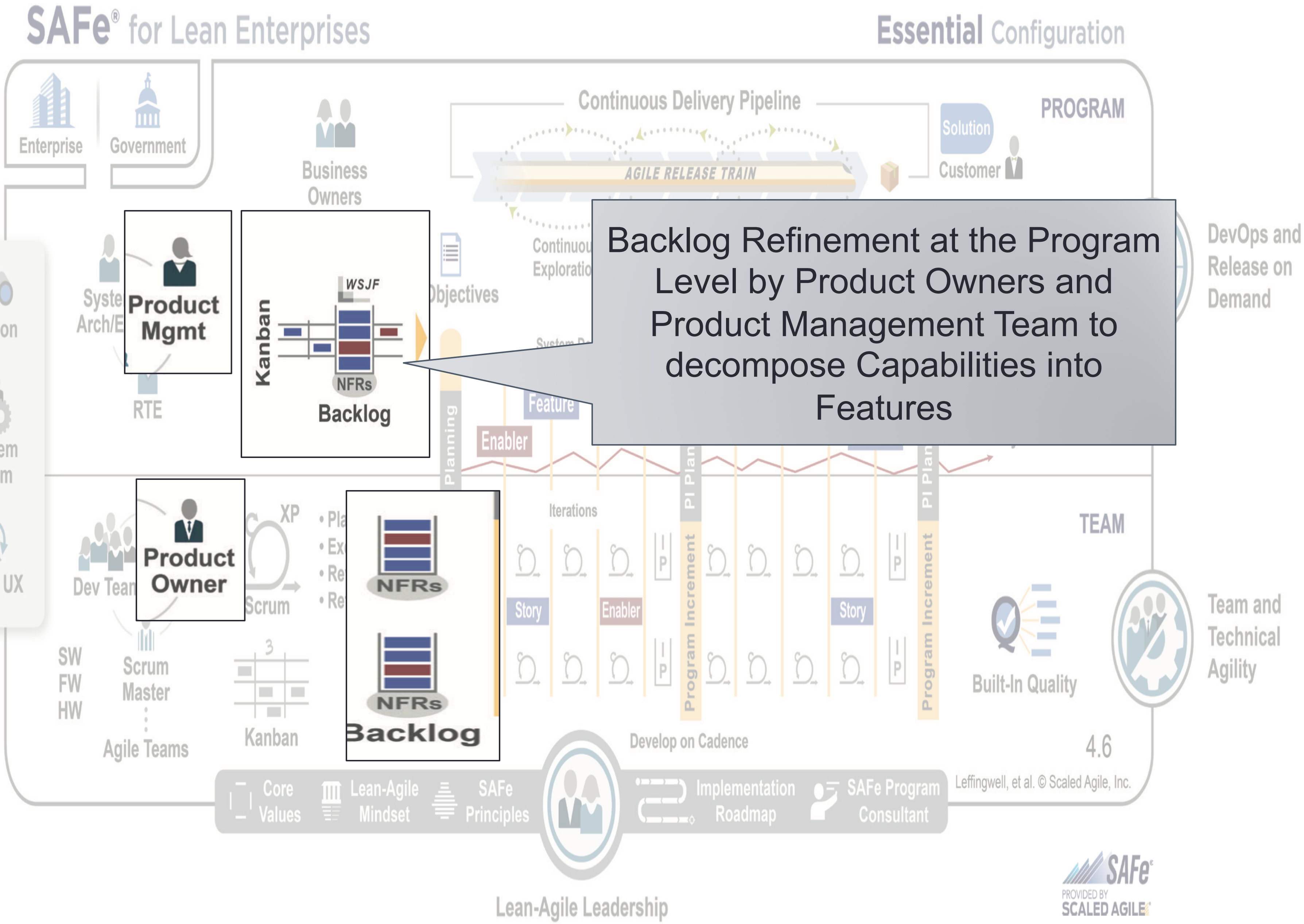
Scaling a broken system won’t work

Common considerations when scaling

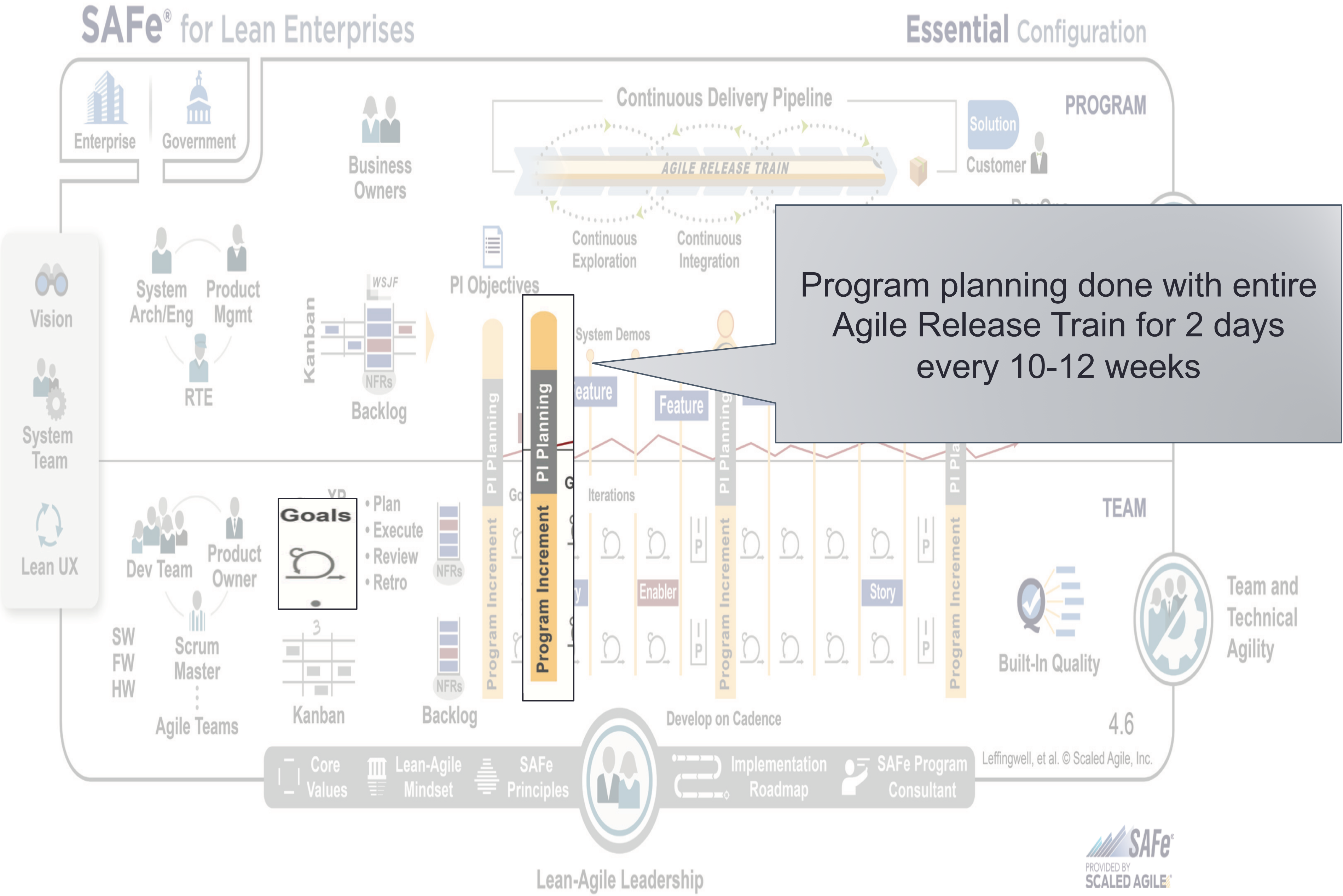
How do we scale:

- Impediment removal?
- Backlog Refinement?
- Program planning?
- Dependencies?


The RTE is the "scaled scrum master" – facilitates events at the program level & escalates impediments



Backlog Refinement at the Program Level by Product Owners and Product Management Team to decompose Capabilities into Features

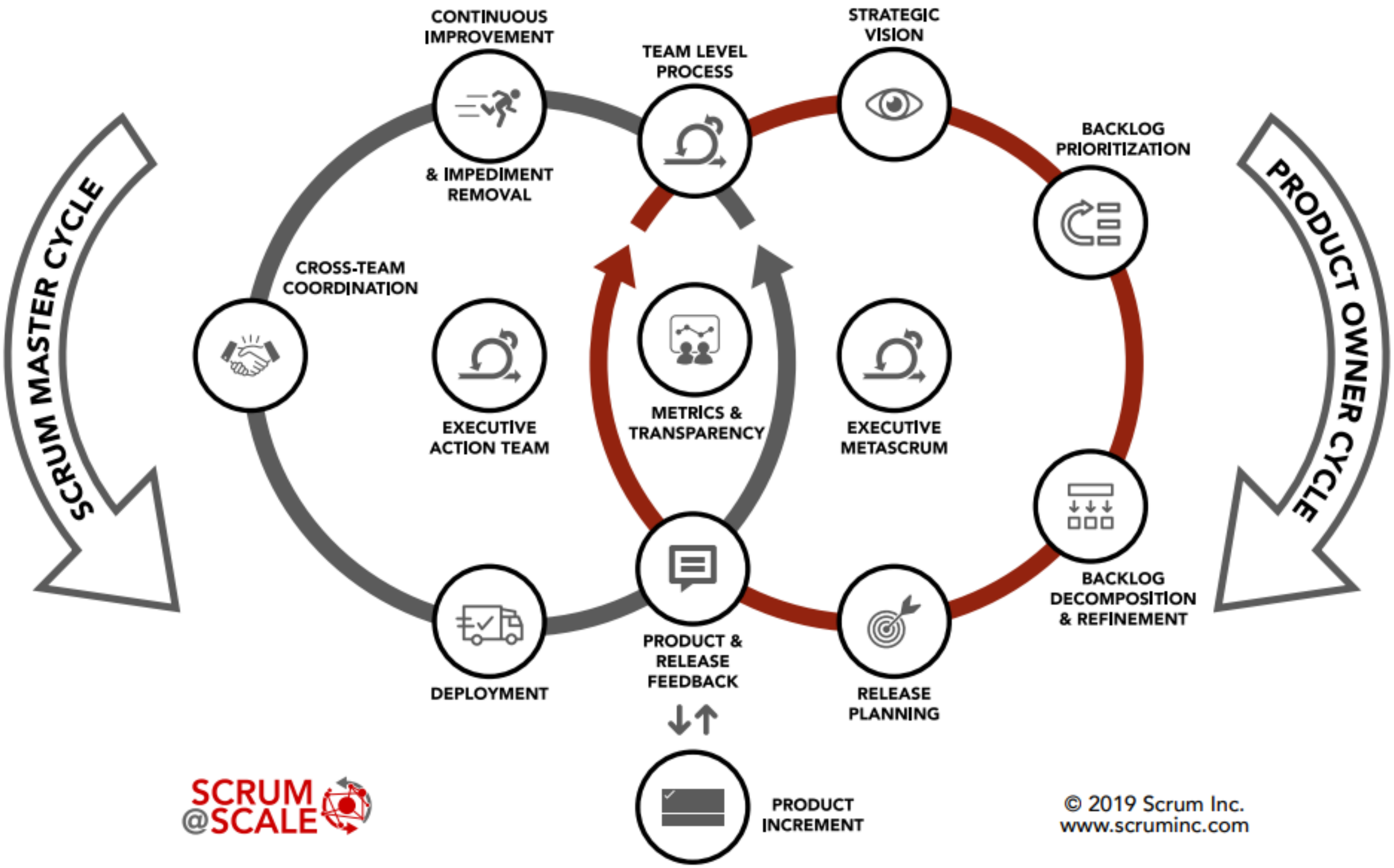


Program planning done with entire Agile Release Train for 2 days every 10-12 weeks

Common issues I’ve seen with SAFe

- 2-day Program Increment Planning is expensive
  - Programs try to save money by shortening
- Synchronizing teams occurs at weekly Scrum of Scrums – escalating impediments often waits for these synchronization points
- Program backlog refinement is not disciplined, so Features are often not well-defined – Program Increment planning is not as effective

Scrum@Scale Framework

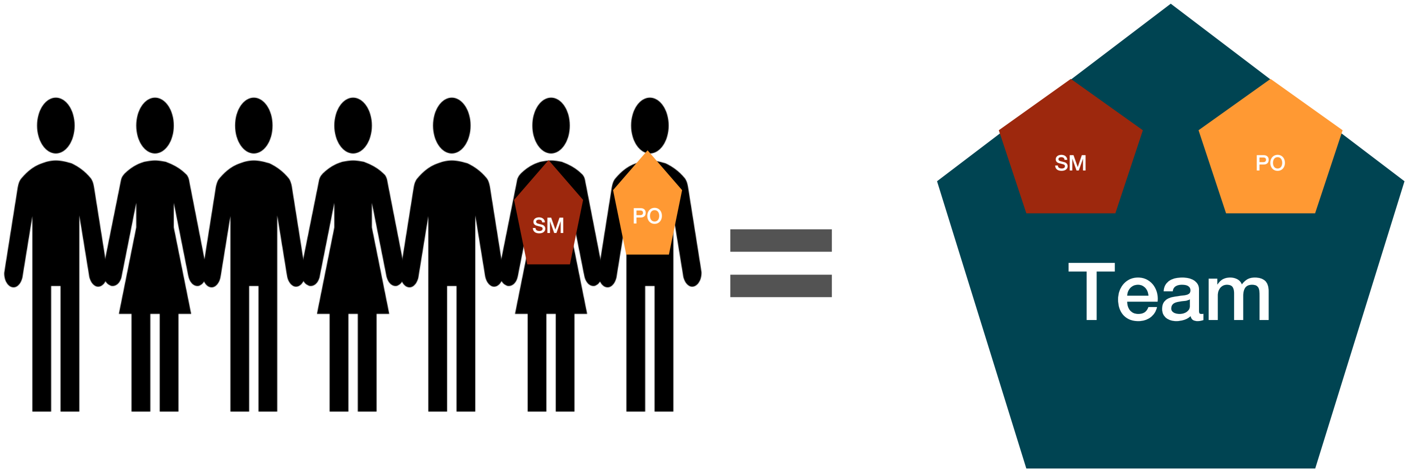


SCRUM @SCALE

Scrum @ Scale

- Jeff Sutherland introduced Scrum@Scale in 1983, but released it officially in February 2018
- S@S is a scale-free architecture
- Just like scrum, S@S is easy to understand, but difficult to master
- Requires committed agile leadership

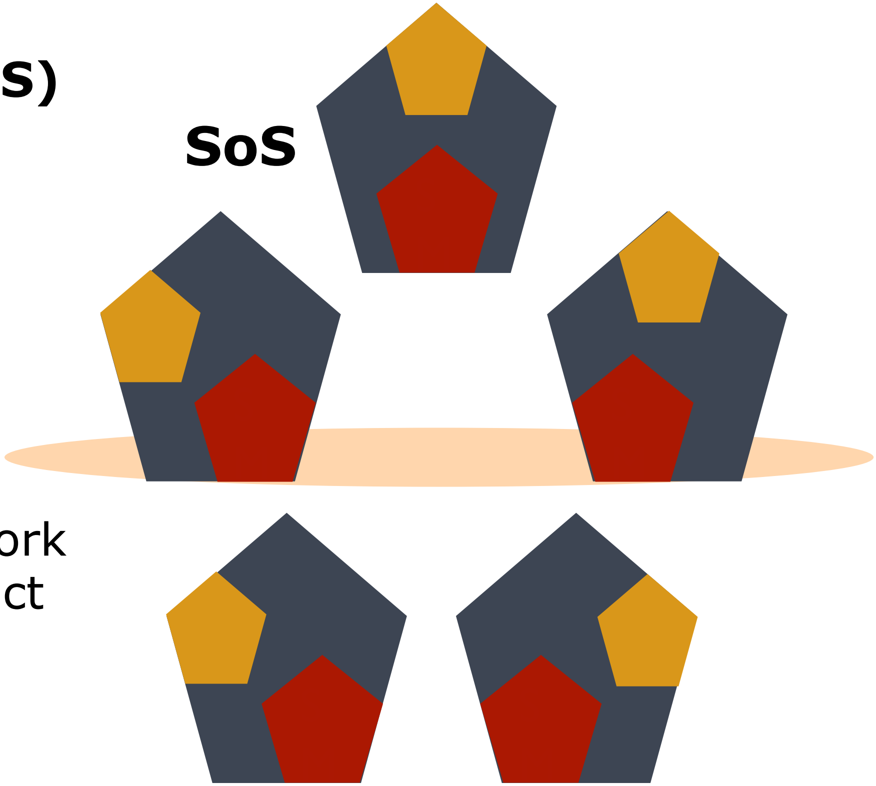
Understanding the Scrum Team Icons



The Scrum of Scrums Perspective

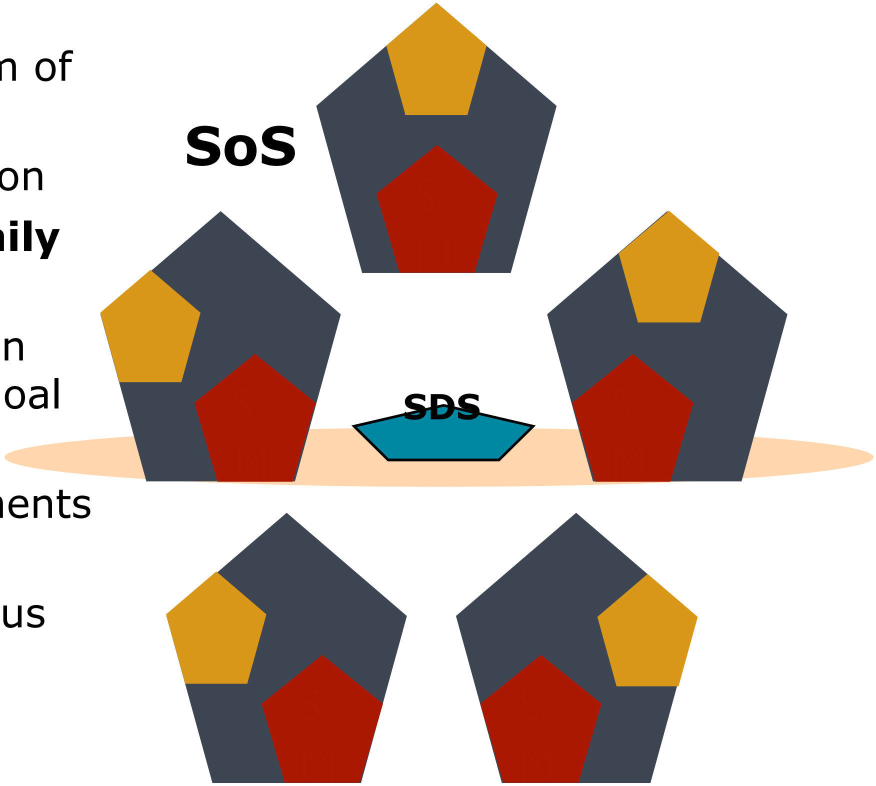
The Scrum of Scrums (SoS) is:

- Not an event
- A “Team of Teams” that have a need to coordinate
- Responsible for a shippable increment of product at the end of a Sprint and must work closely with the Chief Product Owner



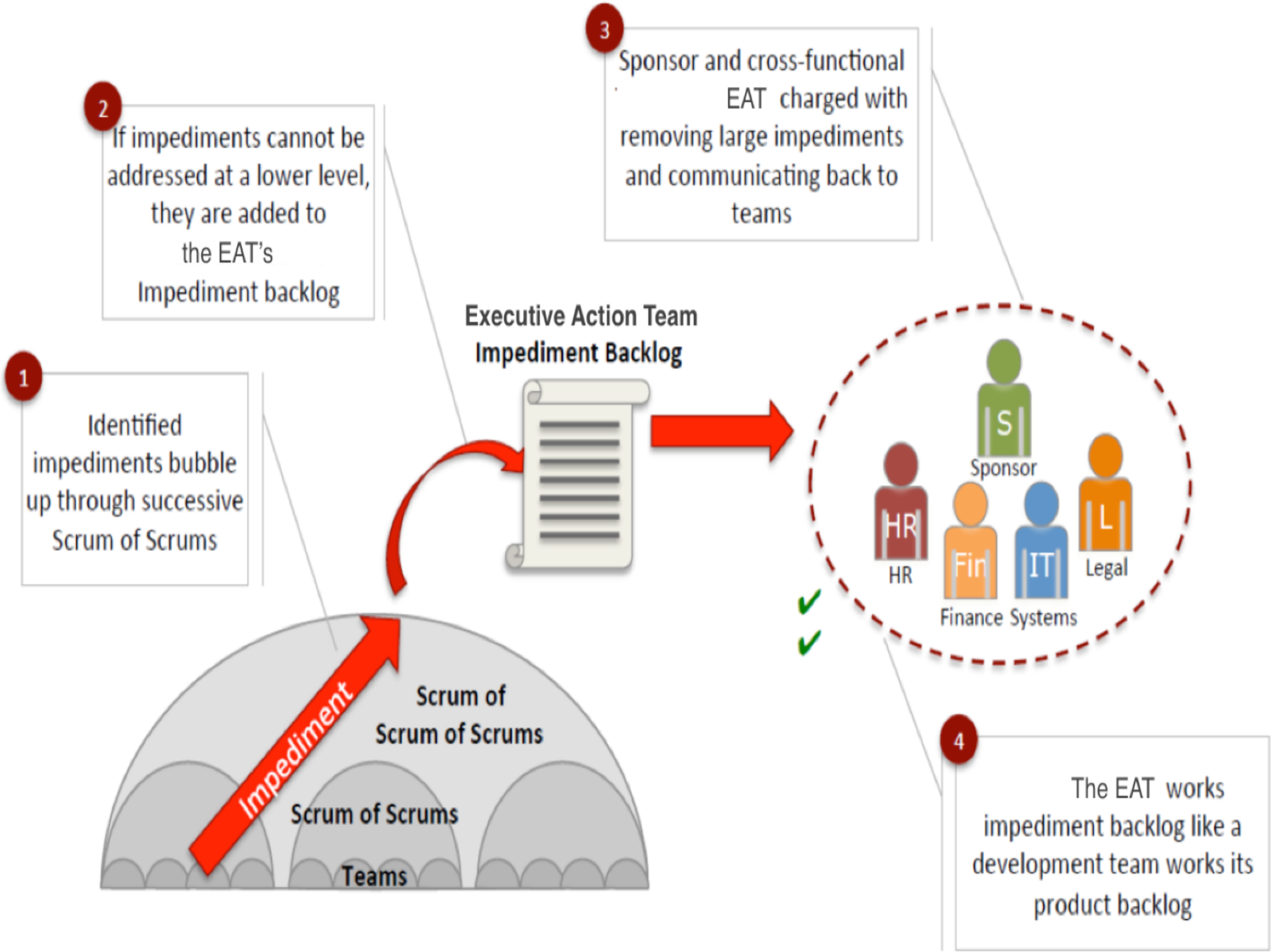
Scaled Daily Scrum (SDS)

- **An event** facilitated by a Scrum of Scrums Master (the SoSM)
- Enables Cross-Team Coordination
- The SDS event **mirrors the Daily Scrum** in that it:
  - is an opportunity to re-plan in order to achieve the Sprint goal for the SoS
  - surfaces & removes impediments if possible
  - Shares learning for Continuous Improvement



Example: 5 teams delivering products together

Typical Impediment Escalation



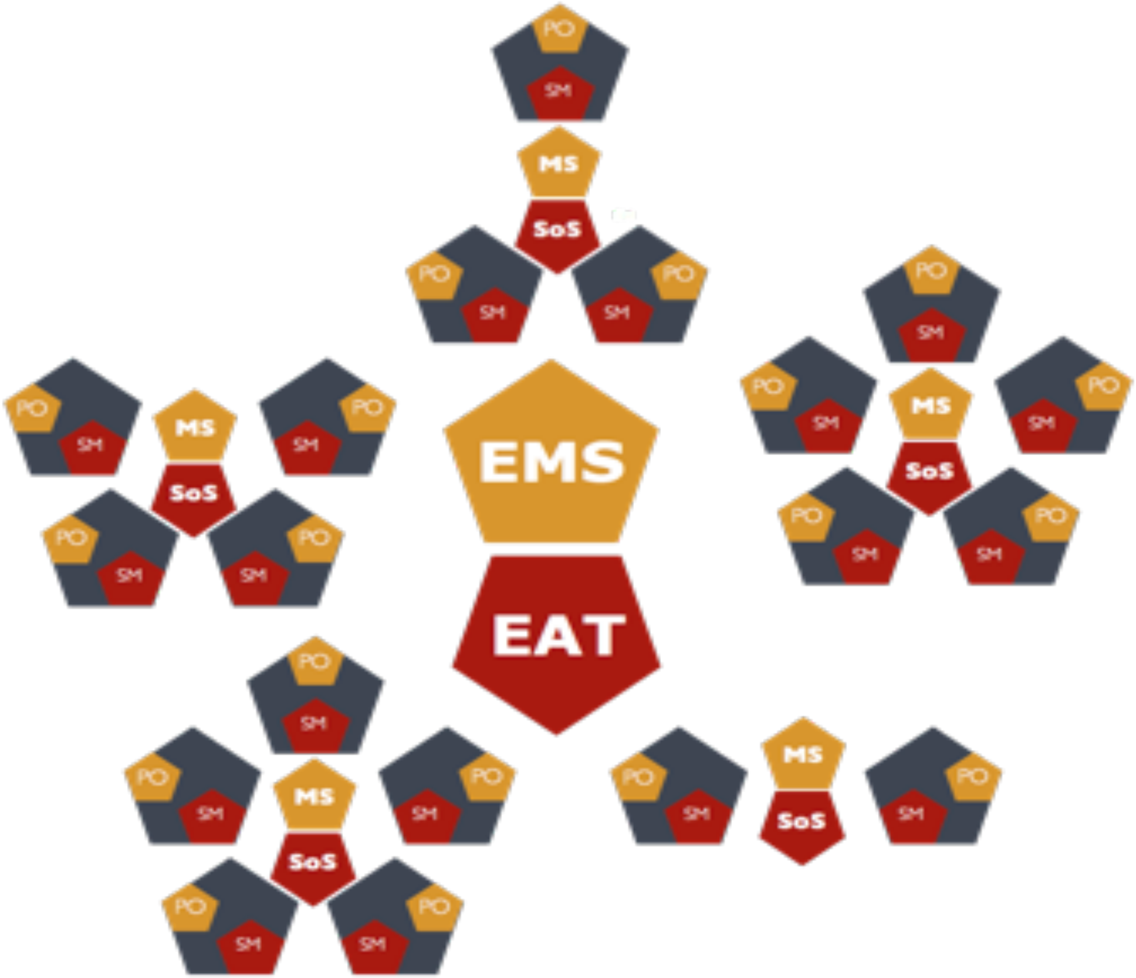
1 Identified impediments bubble up through successive Scrum of Scrums

2 If impediments cannot be addressed at a lower level, they are added to the EAT's Impediment backlog

3 Sponsor and cross-functional EAT charged with removing large impediments and communicating back to teams

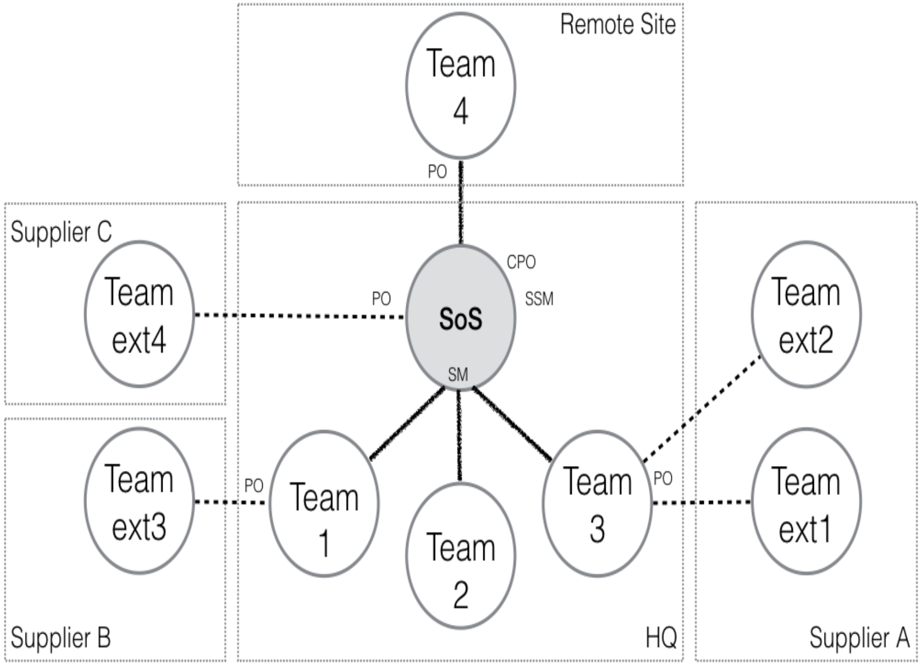
4 The EAT works impediment backlog like a development team works its product backlog

5 SoS's with 19 Teams

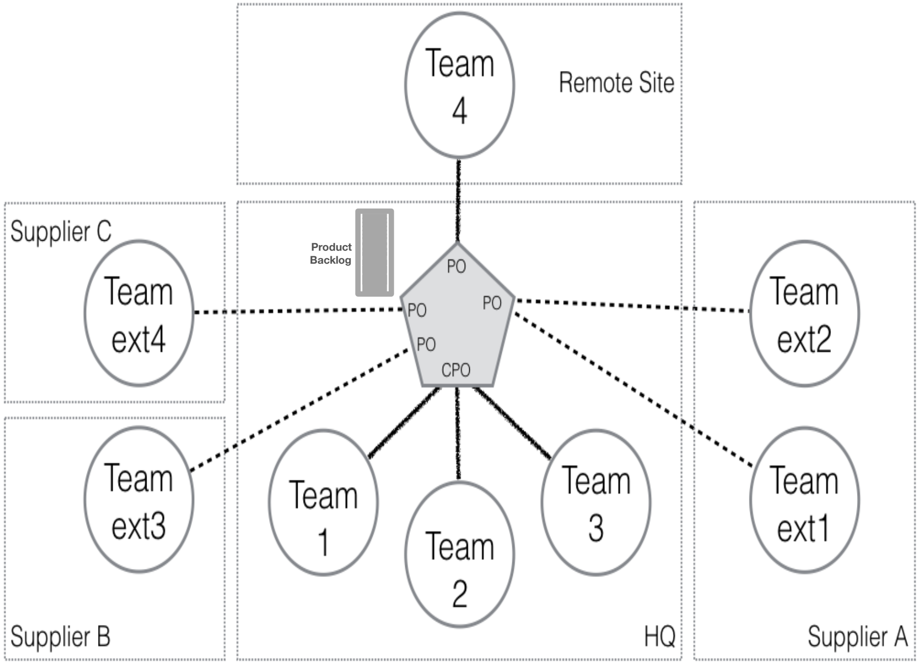


Organizational Structure

Scrum of Scrums



MetaScrum

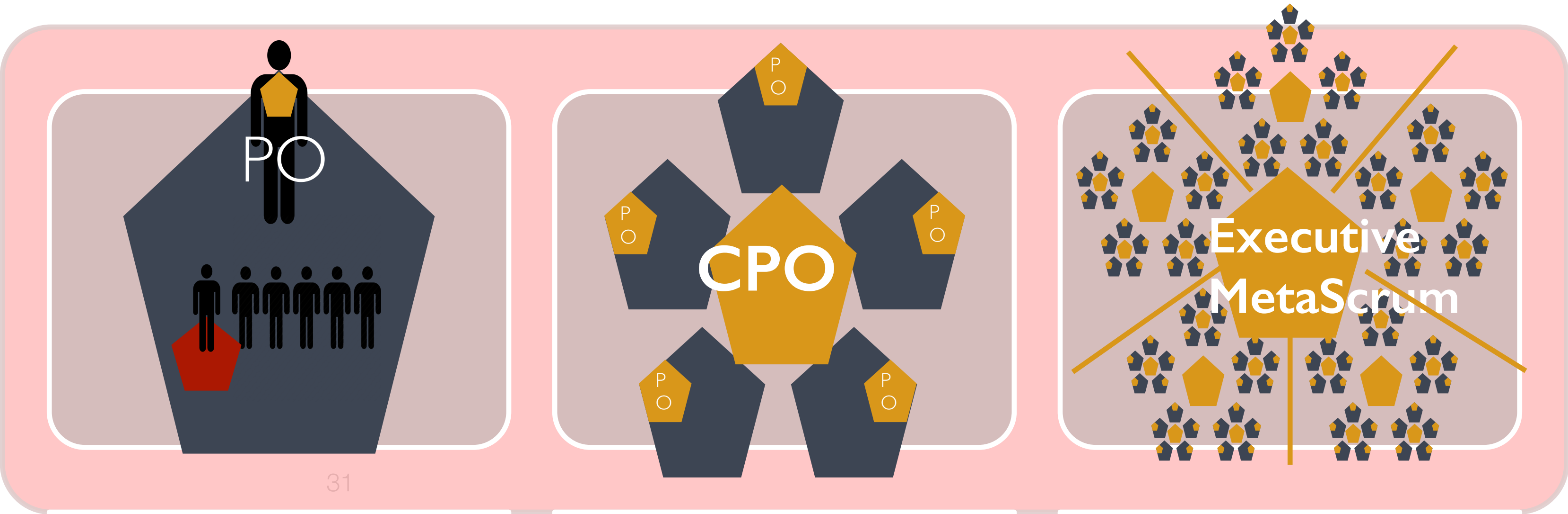


S@S uses Scrum to scale Scrum

Scrum	S@S related concept
Scrum Team	
Product Owner	
Scrum Master	
Sprint Planning	
Daily Scrum	
Sprint Retrospective	
Sprint Review	
Product Backlog	
Sprint Backlog	
Definition of Done	
Impediments List	

S@S
Executive Action Team
Scaled Daily Scrum
Deployment
Metrics & Transparency
Organizational Transformation Backlog
Product & Release Feedback
Scrum of Scrums
Team Backlog
MetaScrum
Release Plan
Chief Product Owner

Scaling the Product Owner



Product Owner

- Team
- Sprints
- Validated Learning

Chief Product Owner

- Multiple Teams
- Roadmap
- Coordinating teams

Chief Chief Product Owner

- Value Streams
- Vision
- Organizational Priorities

The MetaScrum Alignment Meeting

- A MetaScrum Scaled Backlog Refinement Meeting is a gathering of Key Stakeholders, Leadership, Product Owners, and Team members.
- Run by Chief Product Owner
- Did anything change that would change the strategy or priorities?
- Aligns enterprise around single backlog
- *The* forum for stakeholders to express preferences, remove blocks and provide resources needed (they should not try to alter product vision between MetaScrums)
- Produce Owners decompose upcoming backlog